BAT 编程

BAT 文件即批处理文件，是一种包含一条或多条命令无格式的文本文件。其文件扩展名为 bat 或 cmd。在命令提示下键入某个批处理文件的名称或者在窗口界面中双击某个批处理文件后，系统就会调用 cmd.exe 按照该文件中各个命令出现的顺序来顺次运行它们。入侵者常常通过批处理文件的编写来实现多工具的组合入侵、自动入侵及结果提取等功能。

批处理命令简介

在批处理文件中可以按一定顺序将命令进行组合以实现所需的功能，但这种方法往往灵活性不足，而且编写出来的批处理文件也相当烦琐。本节将介绍批处理文件中专用的批处理命令，使用这些命令格式有助于编写高效优质的批处理文件。

1. echo 命令

echo 命令用于打开回显或关闭请求回显功能，也可用于显示消息。如果没有任何参数，echo 命令将显示当前回显设置。

echo 的命令格式为“echo [on/off]/[message]”。其中“/”代表选择，“[]”表示类。这条命令的含义为有 3 种输出方式：echo on；echo off；echo [message]。

实例：echo 命令的使用。

在命令行方式下输入“echo”后会显示如图 11-1。

图 11-1

在命令行方式下输入“echo off”，显示如图 11-2 所示，由于取消了回显功能，命令提示行不再显示。

图 11-2

这时再输入“echo on”，显示如图 11-3 所示，回显功能打开。

图 11-3

输入“echo hello world”，显示如图 11-4 所示，输出信息“hello world”。

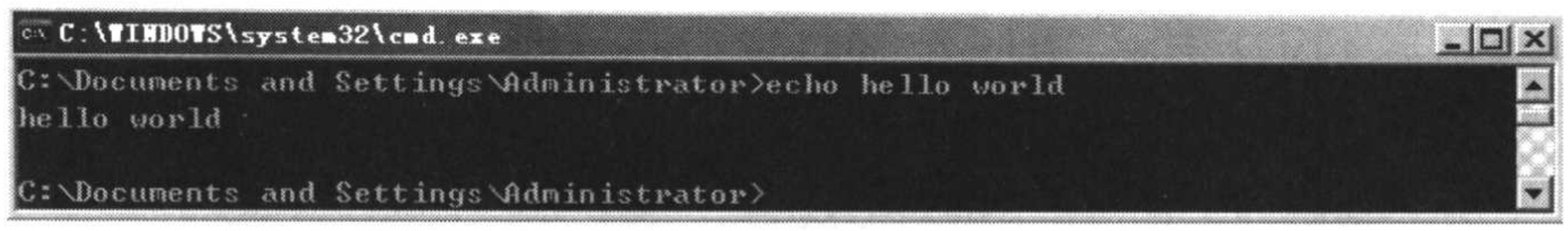

图 11-4

2. @命令

@用于隐藏@后面的命令。批处理文件执行时，在命令行窗口中会显示当前正在运行的命令，为了达到更好的隐蔽性，入侵者可以使用@命令隐藏命令。

实例：@命令的使用。

打开记事本，输入“@echo hello world”，保存为批处理文件，本例中命名为 p.bat。在命令行窗口中运行 p.bat，如图 11-5 所示。

图 11-5

直接得到了命令运行的结果。如果不使用@命令，即在记事本中直接输入“echo hello world”，结果如图 11-6 所示。

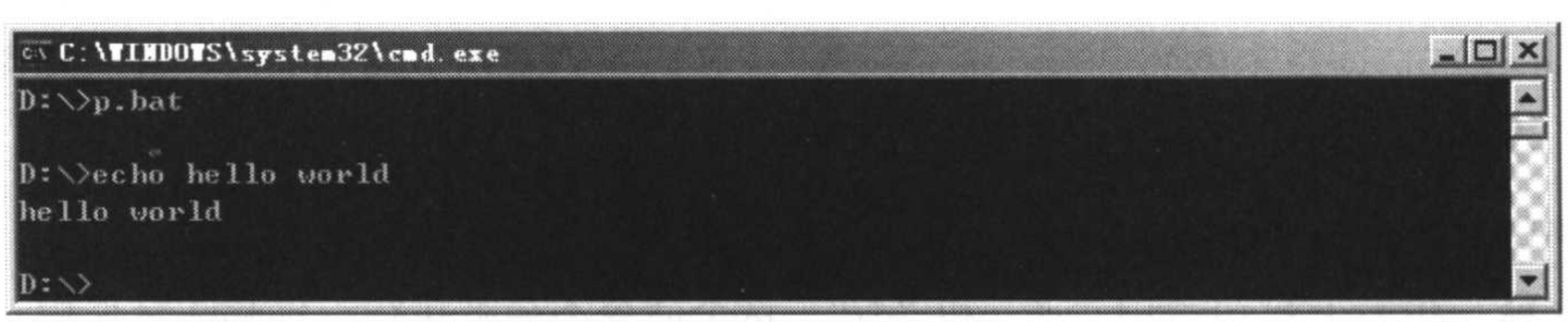

图 11-6

显示了运行时所调用的命令，没有实现信息的隐藏。

3. if 命令

if 是条件语句，判断参数是否符合规定的条件，从而决定执行不同的命令。if 语句有以下 3 种格式。

（1）if "参数" == "字符串"　待执行的命令

参数如果等于指定的字符串，则条件成立，运行命令，否则运行下一句。

（2）if exist 文件名 待执行的命令

如果有指定的文件，则条件成立，运行命令，否则运行下一句。

（3）if errorlevel / if not errorlevel 数字 待执行的命令

如果返回码等于指定的数字，则条件成立，运行命令，否则运行下一句。DOS 程序运行时都会返回一个数字给 DOS，称为错误码（errorlevel）或返回码，常见的返回码为 0、1。

实例：if 命令的使用。

在前面已生成了一个 p.bat。打开记事本输入“@if exist p.bat echo succeed”，保存为 if.bat。在命令行窗口中运行 if.bat，如图 11-7 所示。

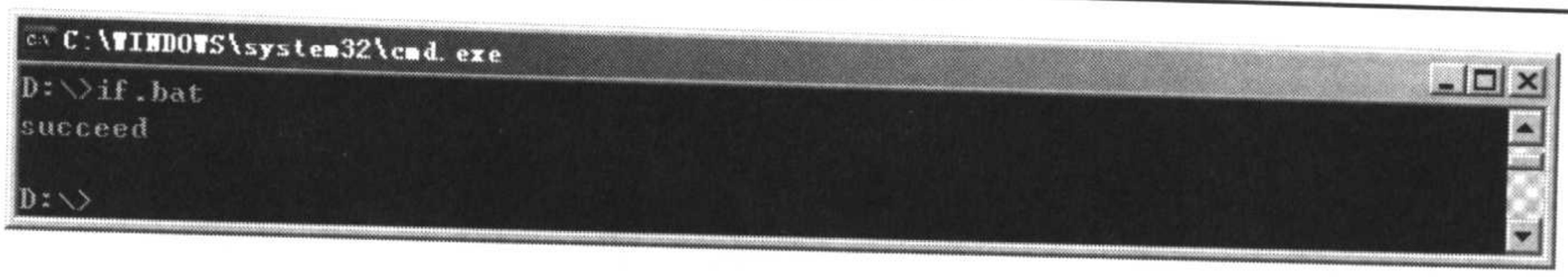

图 11-7

4. goto 命令

用于跳转到标签指定的位置并从标签的下一行命令开始执行。命令格式为“goto label”，其中 label 为标签。标签的名字任意，但是最好是有意义的字母。在字母前需要加“:”号用来表示这个字母是标签。

实例：goto 命令的使用。

打开记事本，输入如下的命令。

```
:dv @if exist p.bat echo succeed
@goto dv
```

保存为 goto.bat，批处理文件运行如图 11-8 所示。

图 11-8

goto.bat 运行后，会不停的显示“succeed”，这是由于没有结束循环的语句导致程序始终处于循环状态，本例的目的只是展示 goto 命令的作用。

5. rem 命令

用于注释，相当于 C 语言中“/*”和“*/”，这个命令不会被执行，仅起注释作用。

rem 命令的命令格式为“rem message”。

6. pause 命令

挂起命令，当批处理文件运行到 pause 命令时，会出现“请按任意键继续. . .”的提示，如图 11-9 所示。

图 11-9

7. call 命令

用于在一个批处理程序中调用另一个批处理程序，此调用过程中只是暂时挂起父批处理程序，当子批处理程序运行完毕后会返回父批处理程序继续执行。

常用的命令格式为“call [path] filename”，其中 path 是要调用的批处理文件的位置，可选，默认是与父批处理程序在同一目录下；filename 用于指定子批处理程序的文件名，其必须包含有.cmd 或.bat 后缀名。

事实上 call 命令的格式不止如此，但不常用，详细情况可以在命令行窗口中输入 call /?查看。

实例：call 命令的使用。

首先在记事本中输入“if exist p.bat @echo succeed”，保存为 p.bat。

然后新打开一个记事本输入“@call p.bat”，保存为 call.bat。这里将 call.bat 和 p.bat 保存在了同一个目录下。在命令行窗口中运行 call.bat，如图 11-10 所示。

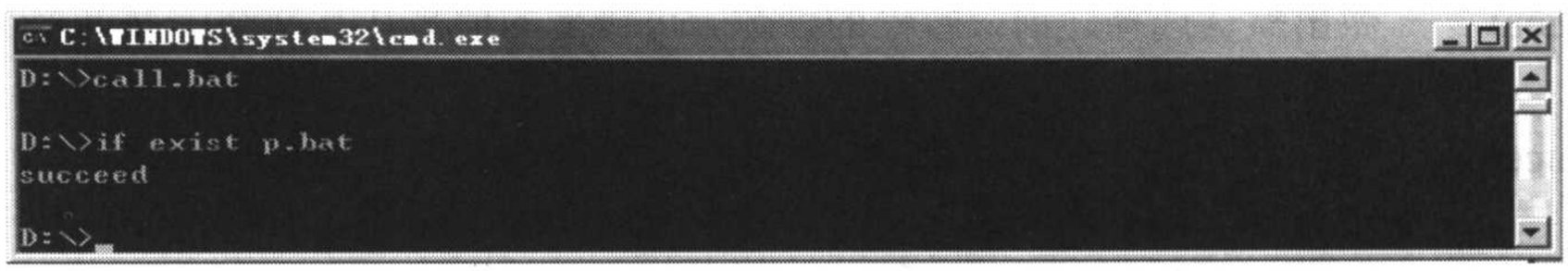

图 11-10

8. start 命令

用于外部命令的调用，所用的 DOS 命令和命令行程序都可以由 start 命令调用。

start 命令的调用格式为“start ["title"] [/D path] [/I] [/MIN] [/MAX] [/SEPARATE | /SHARED] [/LOW | /NORMAL | /HIGH | /REALTIME | /ABOVENORMAL | /BELOWNORMAL] [/WAIT] [/B] [command/program] [parameters]”。

下面仅对常用参数进行介绍：

MIN	开始时窗口最小化。
SEPARATE	在分开的空间内开始 16 位 Windows 程序。
HIGH	在 HIGH 优先级类别开始应用程序。
REALTIME	在 REALTIME 优先级类别开始应用程序。
WAIT	启动应用程序并等候它结束。
parameters	这些为传送到命令 / 程序的参数。

实例：start 命令的使用。

打开记事本输入“start p.bat”，保存为 start.bat，在命令行窗口中运行后如图 11-11 所示。

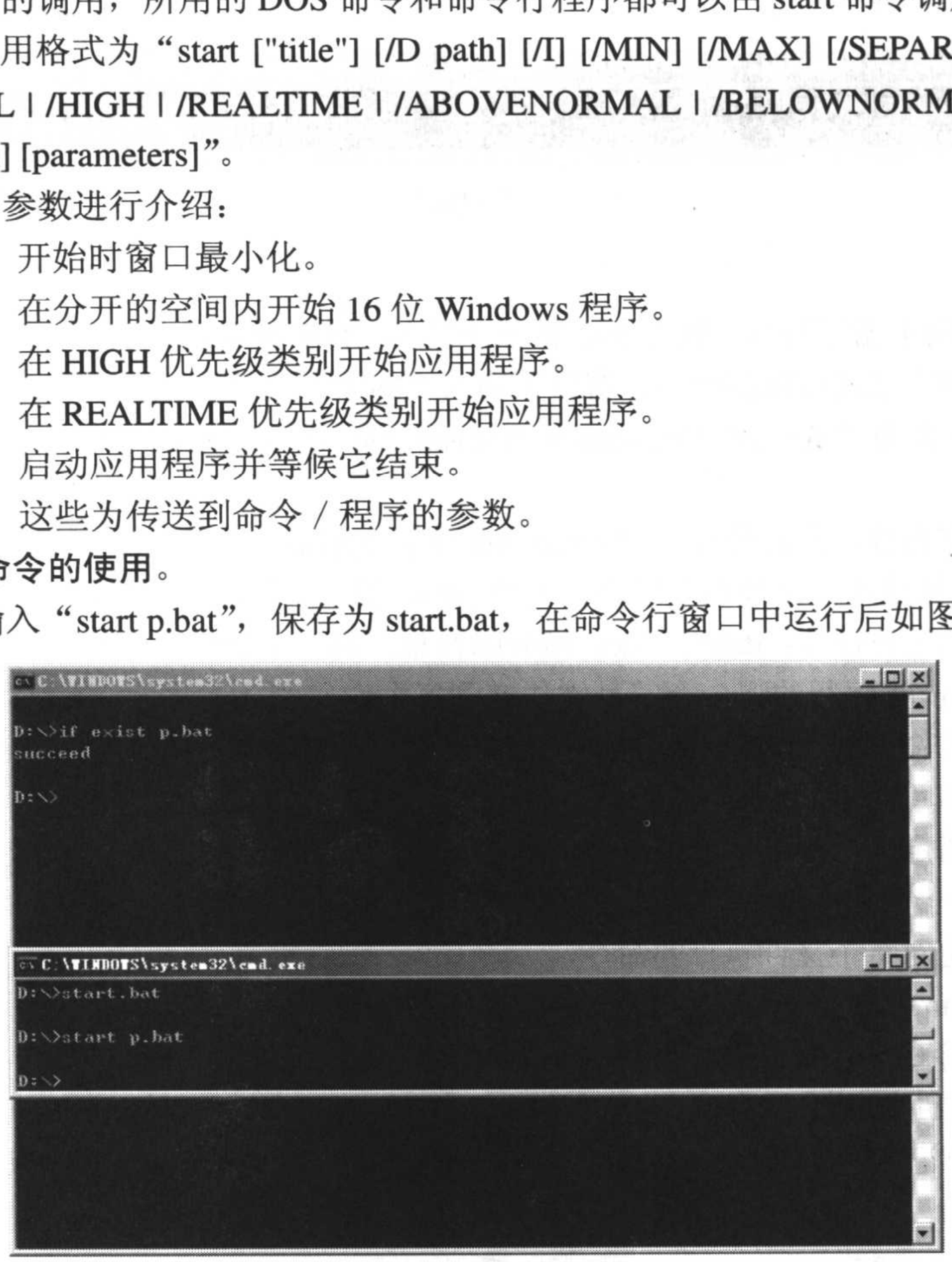

图 11-11

start 运行后会在一个新的命令行窗口中调用 p.bat。

9. choice 命令

choice 命令可以让用户输入一个字符，从而运行不同的命令，其命令格式为“choice [/C choices] [/N] [/CS] [/T timeout /D choice] [/M text]”。各参数含义如下：

/C choices	指定要创建的选项列表。默认列表是"YN"。

/N	在提示符中隐藏选项列表。提示前面的消息得到显示选项依旧处于启用状态。
/CS	允许选择分大小写的选项。在默认情况下，这个工具是不分大小写的。
/T timeout	做出默认选择之前，暂停的秒数。可接受的值是从 0~9999。如果指定了 0，就不会有暂停，默认选项会得到选择。
/D choice	在 *n* 秒之后指定默认选项。字符必须在用 /C 选项指定的一组选择中；同时，必须用/T 指定 *n*。
/M text	指定提示之前要显示的消息。如果没有指定，工具只显示提示。

实例：choice 命令的使用。

在命令行窗口中输入“choice /C YNC /M"确认请按 Y，否请按 N，或者取消请按 C。""，运行后显示如图 11-12 所示。

图 11-12

10. for 命令

for 命令主要用于循环调用，该命令功能十分强大，有多种调用方法，这里只介绍一种常用命令格式，其他的调用格式可以在命令行窗口下输入“for /?”查看。

常用命令格式为“for /F ["options"] %%variable IN (file-set) DO command [command-parameters]”。

/F 表示跳过空白行，同时分行执行 command 中指定的命令。

"options"用于指定不同功能的关键字，其中 eol=c 指一个行注释字符的结尾；skip=n，用于指定在文件开始时跳过的行数；delims=xxx 指分隔符集，是一个替换了空格和跳格键的默认分隔符集；tokens=x,y,m-n 指每行的哪一个符号被传递到每个迭代的 for 本身，m-n 格式为一个范围；usebackq 指定允许命令执行一个后引号的字符串并且一个单引号字符为文字字符串命令并允许在（file-set）中使用双引号括起文件名称。

%%variable 指定一个单一字母可替换的参数。

(file-set)指定一个或一组文件，可以使用通配符。

command 指定对每个文件执行的命令。

command-parameters 为特定命令指定参数或命令行开关。

实例：for 命令的使用。

打开记事本文件依次输入 1~5，注意每输入一个数字换一行，保存为 zl.txt。

重新打开一个记事本文件输入“@for /f %%i in (zl.txt) do @echo %%i”，保存为 for.bat，运行后如图 11-13 所示。

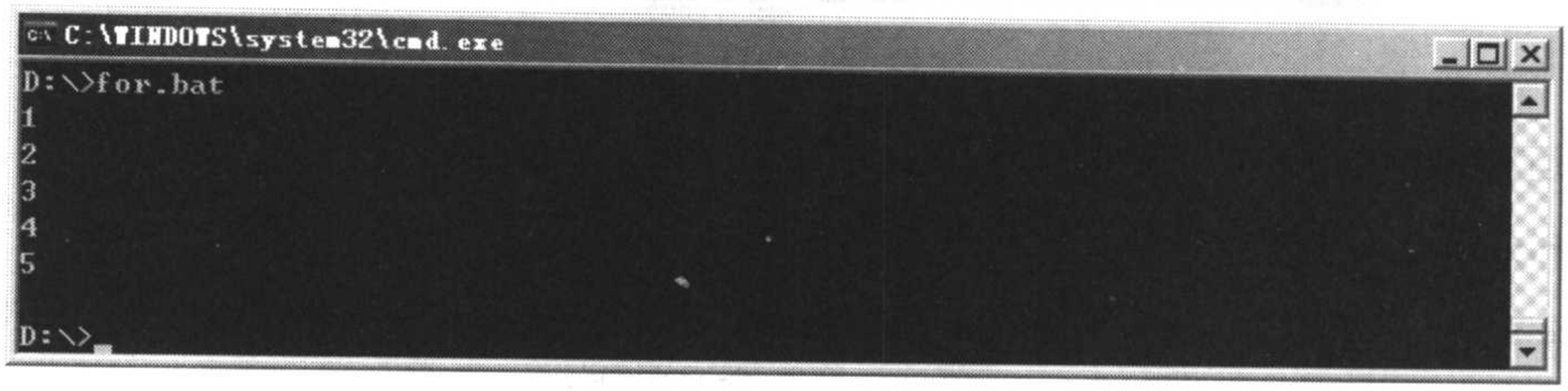

图 11-13

11. del 命令

del 命令用于删除一个或多个文件。其命令格式为“del [/P] [/F] [/S] [/Q] [/A[[:]attributes]] names”。其中 P 用于删除每一个文件之前提示确认；F 用于强制删除只读文件；S 用于从所有子目录删除指定文件；Q 为安静模式，删除全局通配符时，不要求确认；A 用于根据属性选择要删除的文件；attributes 用于对应 A 所要删除文件的属性，包括的参数值中 R 代表只读文件，S 代表系统文件，H 代表隐藏文件，A 代表存档文件，-表示“否”的前缀；names 用于指定一个文件名或者一个目录，指定目录时，其中所包含的所有文件都会被删除。

例如删除，p.bat 文件，在命令行窗口中输入“del f p.bat”，运行后如图 11-14 所示。

图 11-14

12. net 命令

这里只介绍开启网络服务和关闭网络服务的命令格式。

“net start [service]”用于开启网络服务，service 用于指定需要开启的网络服务名，若不指定则开启所有能够开启的网络服务。

“net stop service”用于关闭网络服务，其中 service 是需要关闭的网络服务名，此项必须指定。

以上介绍了入侵者常用的一些命令，需要说明的是这些命令并不是只能在批处理文件中使用。所谓批处理文件实际上是不同功能命令的集合。

11.2　在批处理文件中使用参数与组合命令

11.2.1　在批处理文件中使用参数

批处理中可以使用参数，一般从%0 到%9。若想使用超过%10 及其后的参数，需要使用 shift 命令进行移动，由于这种情况比较罕见，本节不做介绍。

举例来说明参数的含义与使用方法。

实例：参数的使用。

打开记事本文件输入，如下的代码：

```
if "%1"=="s1" echo this is %1
if "%2"=="s2" echo this is %2
```

保存为 c.bat。在命令行窗口中输入“c.bat s1 s2”，如图 11-15 所示。

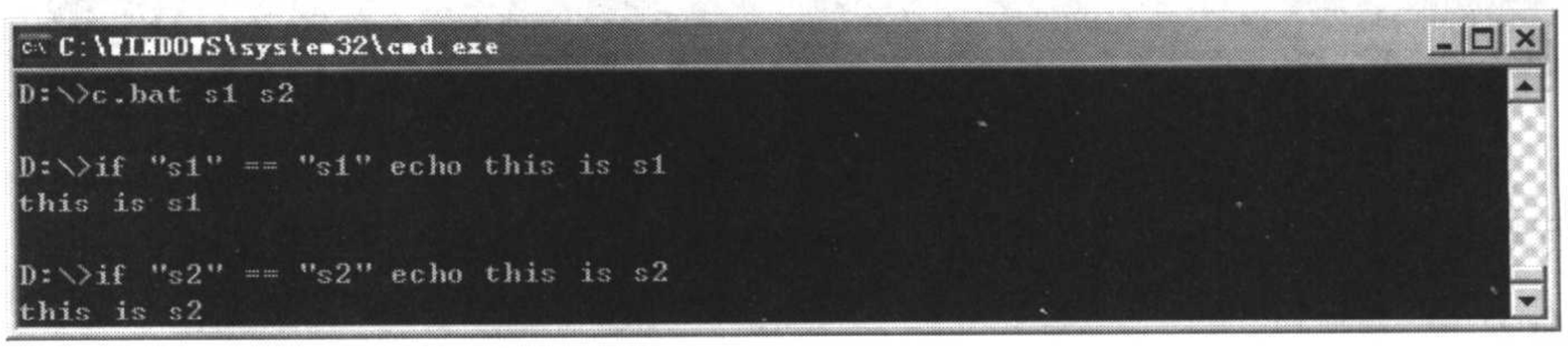

图 11-15

在实例中，“c.bat”运行时带有两个参数“s1”和“s2”，程序运行后，从结果来看%1 的值为 s1，%2 的值为 s2，即实现了参数的替换。此外，%0 用来代表 c.bat 本身。

参数的一个经常用法是用于判断，根据给出的参数，批处理文件在其内部命令中进行判断，从而确定下一步该如何操作。

11.2.2 组合命令

1. &命令

使用格式为“第一条命令&第二条命令&第三条命令……”。使用&组合命令可以同时执行多条命令，而不管命令是否执行成功。

可以查看一下&命令的运行效果，打开命令行窗口输入“echo 1 & echo 2 & echo3”，运行后如图 11-16 所示。

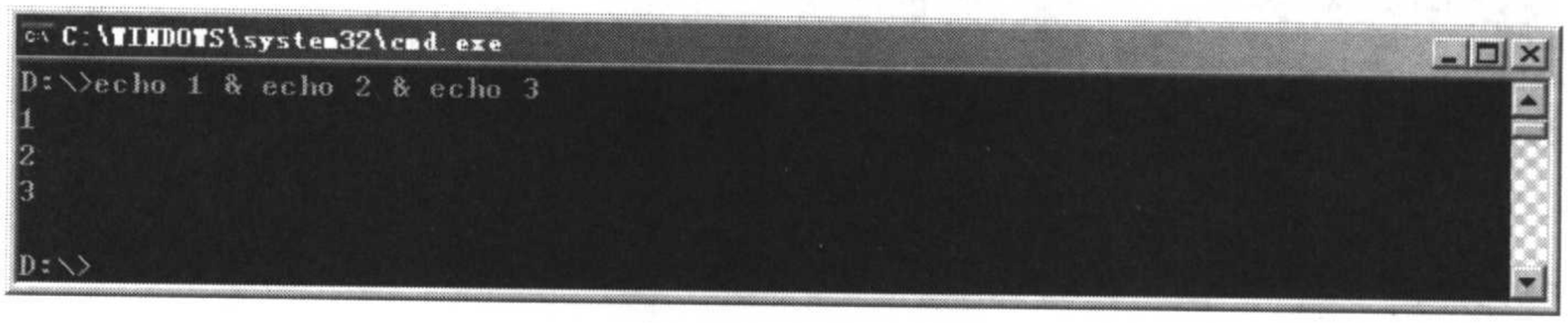

图 11-16

2. &&命令

使用格式为“第一条命令 && 第二条命令&& 第三条命令……”。使用&&组合命令也可以同时执行多条命令，但如果其中的某一条命令执行失败，则组合命令执行终止。

举例来具体说明，打开记事本文件输入如下代码：

```
echo 1 && echo 2 && if "%1"=="s1" echo 3 && echo 4
```

保存为 g.bat。在命令行窗口中输入“g.bat s1”，运行结果如图 11-17 所示。

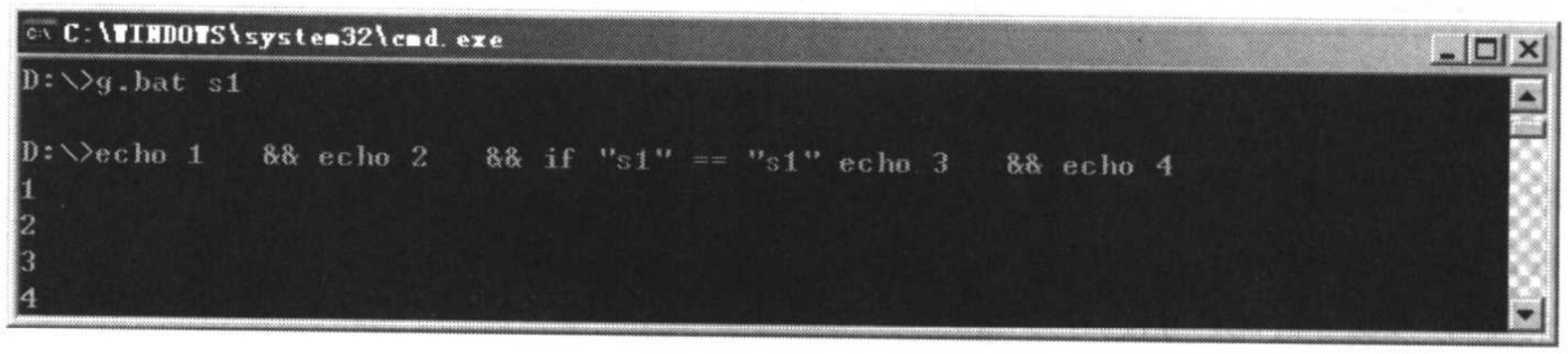

图 11-17

在命令行窗口中输入“g.bat s”，运行结果如图 11-18 所示。

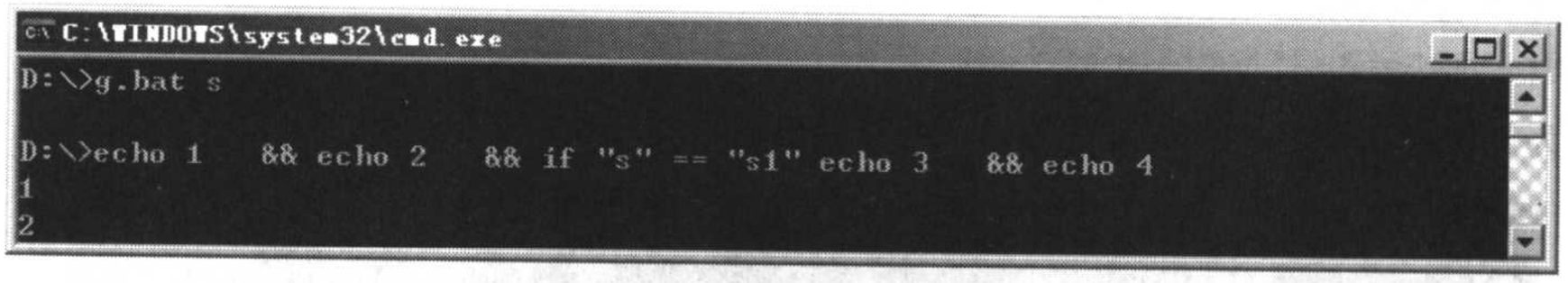

图 11-18

从运行结果可以看出&&构成的组合命令与&构成的组合命令之间的差别。

3. ||命令

使用格式为“第一条命令 || 第二条命令|| 第三条命令……”。使用||可以同时执行多条命令，当碰到执行正确的命令后将不执行后面的命令，如果没有出现正确的命令则一直执行完所有命令。

实例：||命令的使用。

修改 g.bat 文件，将其内容修改为“if "%1"=="s1" echo 1 || echo 2 || echo 3 || echo 4”。打开命令行窗口，输入“g.bat s1”，运行后如图 11-19 所示。

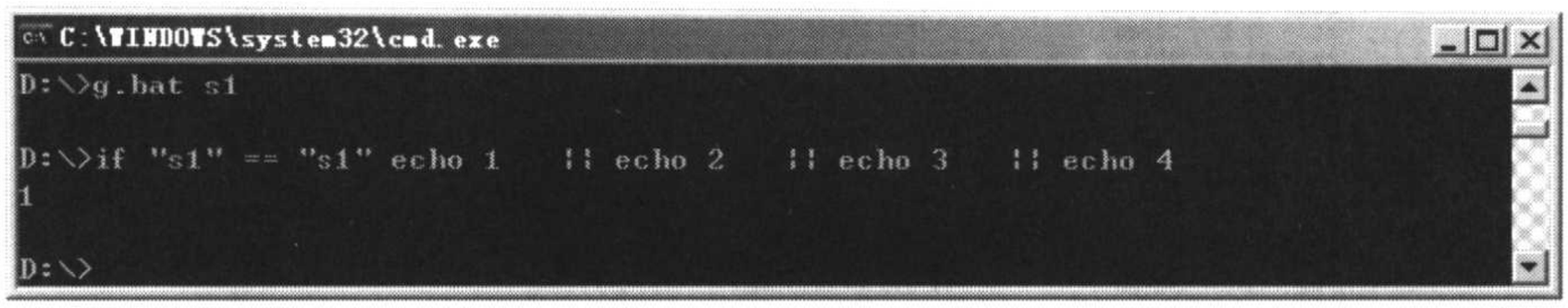

图 11-19

11.3　管道命令

1. |命令

命令使用格式为“第一条命令 | 第二条命令| 第三条命令……”，其含义是将前一条命令的结果作为后一条命令的参数来使用。

2. >、>>输出重定向命令

这两条命令用于将一条命令或某个程序输出结果重定向到指定文件中，>与>>的区别在于>会清除原有文件中的内容后写入指定文件，而>>只追加内容到指定文件中，不删除原有文件的内容。

一个简单的实例是“echo hello world>d:\hello.txt”，运行后会将“hello world”写入 hello.txt 中，如果 hello.txt 不存在则会创建该文件。

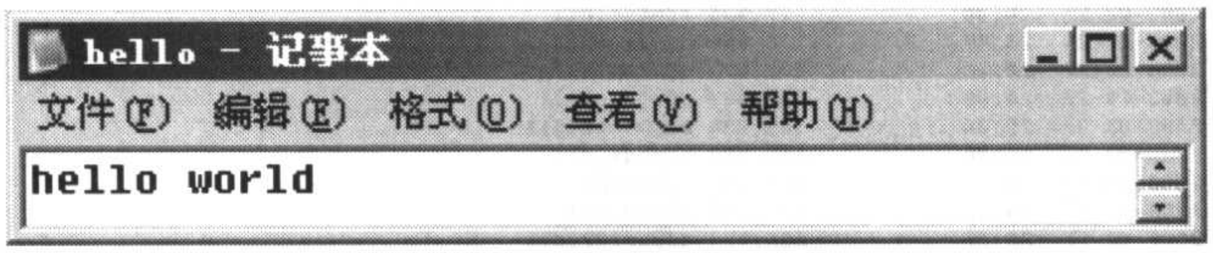

图 11-20

3. <、>&、<&命令

<命令用于从文件中而不是从键盘中读入命令输入。

>&命令用于将一个句柄的输出写入到另一个句柄的输入中。

<&命令用于从一个句柄读取输入并将其写入到另一个句柄输出中。

以上的三条管道命令并不常用，这里不做更多的介绍。

4. 管道命令的使用实例——手动查找 system32 目录中可能存在的木马

思路：当怀疑本机中了木马时，即刻导出 system32 目录下的 exe 文件和 dll 文件。与正常状态下的备份进行比较，找出多出来的文件，再在这些文件中进一步确定是否存在木马。

步骤 1　正常状态下的备份

在命令行窗口中转到 system32 目录下，输入“dir *.exe>d:\exeb.txt & dir *.dll>d:\dllb.txt”，运行后，如图 11-21 到图 11-23 所示。

图 11-21

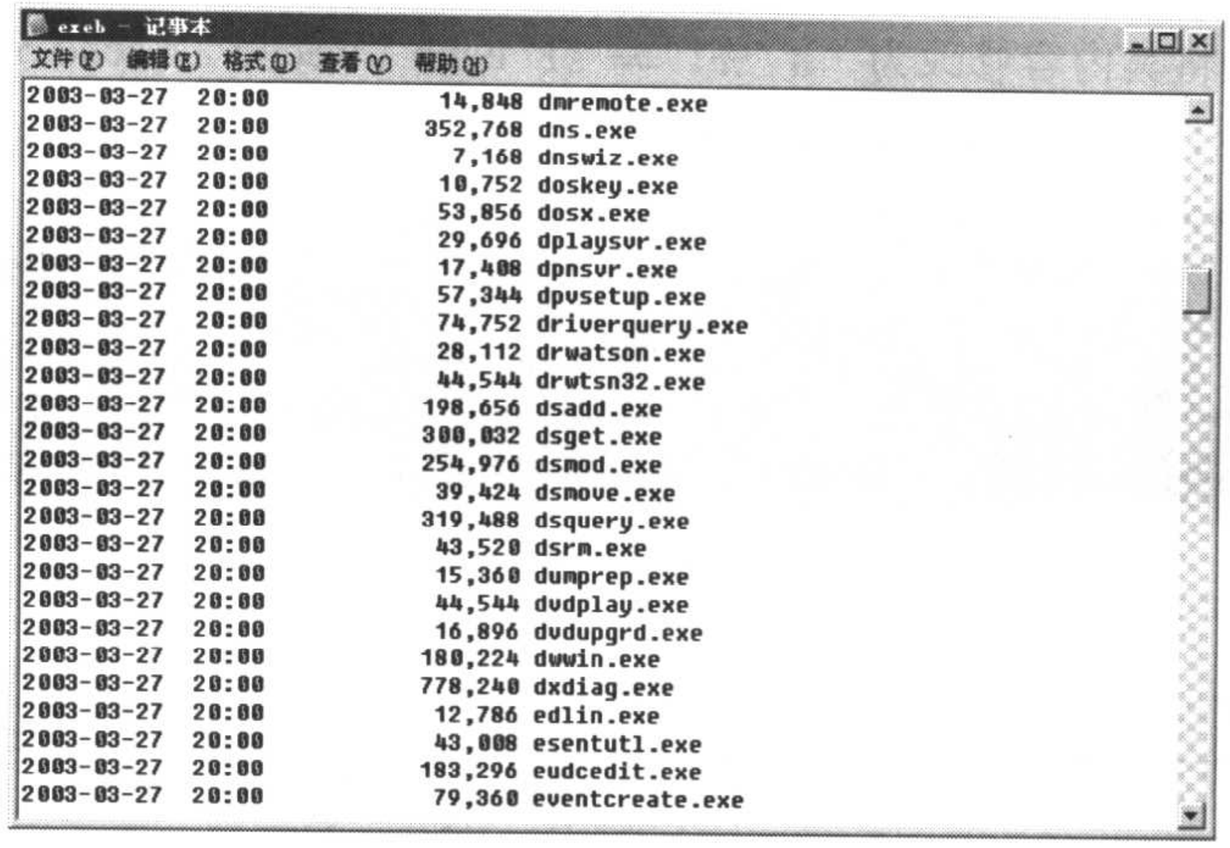

图 11-22

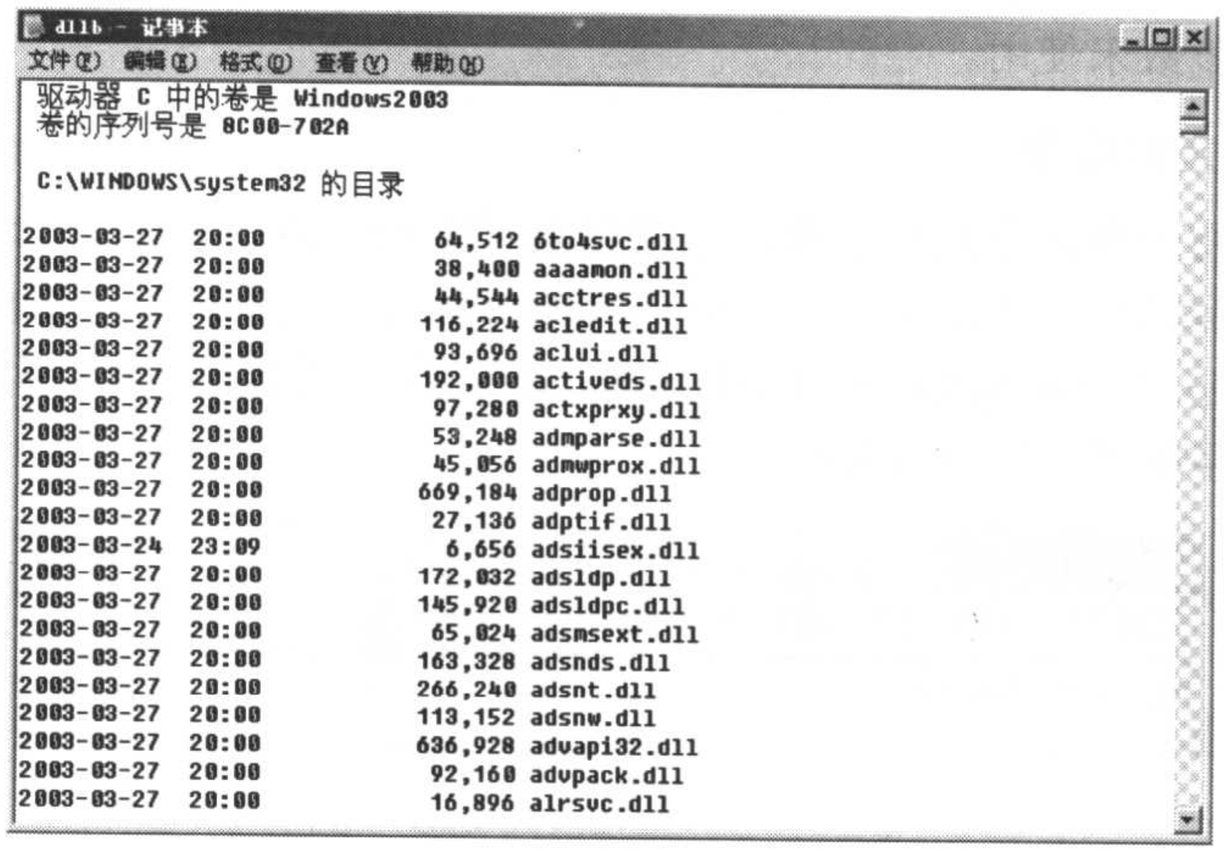

图 11-23

步骤 2 导出异常时的 exe 文件列表与 dll 文件列表

假定导出了 exec.txt 和 dllc.txt，并保存在 D 盘根目录下。

具体过程略。

步骤 3 比较前后两次的文件列表

在命令行窗口中输入“fc d:\exeb.txt d:\exec.txt>>d:\diff.txt & fc d:\dllb.txt d:\dllc.txt>> d:\diff.txt”，运行后如图 11-24 和图 11-25 所示。

图 11-24

步骤 4 确定新增文件是否是木马文件

（略）。

为了简便操作，可以将本例中所用到的相关命令写入批处理文件中。

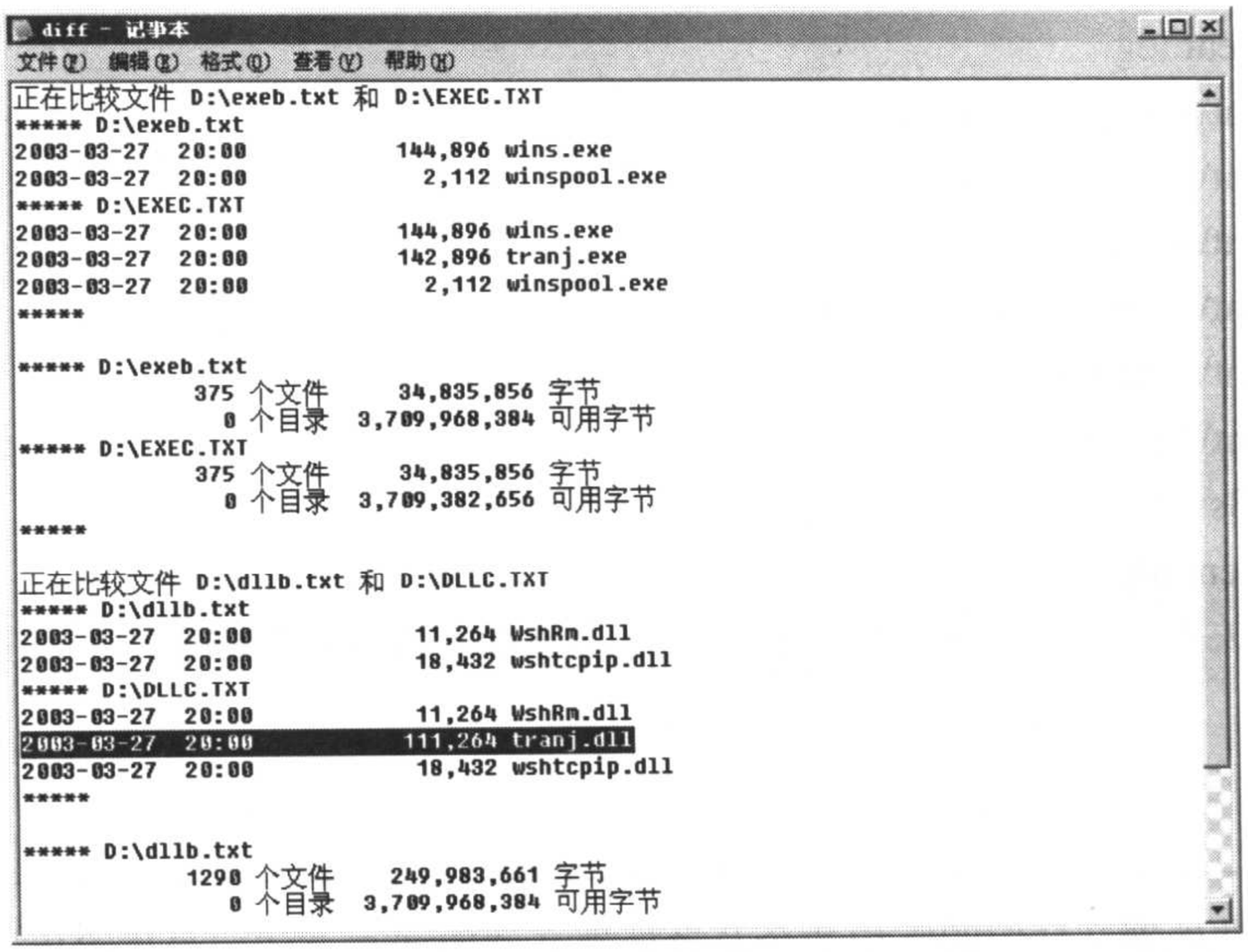

图 11-25

11.4　综合利用的实例

本章的前三节着重介绍了批处理文件所用到的命令，在本节中将介绍几个综合利用的实例。

11.4.1　系统加固

通过批处理文件实现系统加固，即关闭一些不必要的服务和功能，也可用于给“肉鸡”打补丁，增加“肉鸡”的安全性。

图 11-26 中包含了实现上述功能的简易代码，代码中只完成了最基本的保护措施，在完成一个功能后，有相应功能的说明。这段代码中用到了注册表选项，有关注册表的内容可以参考一些相关的资料。

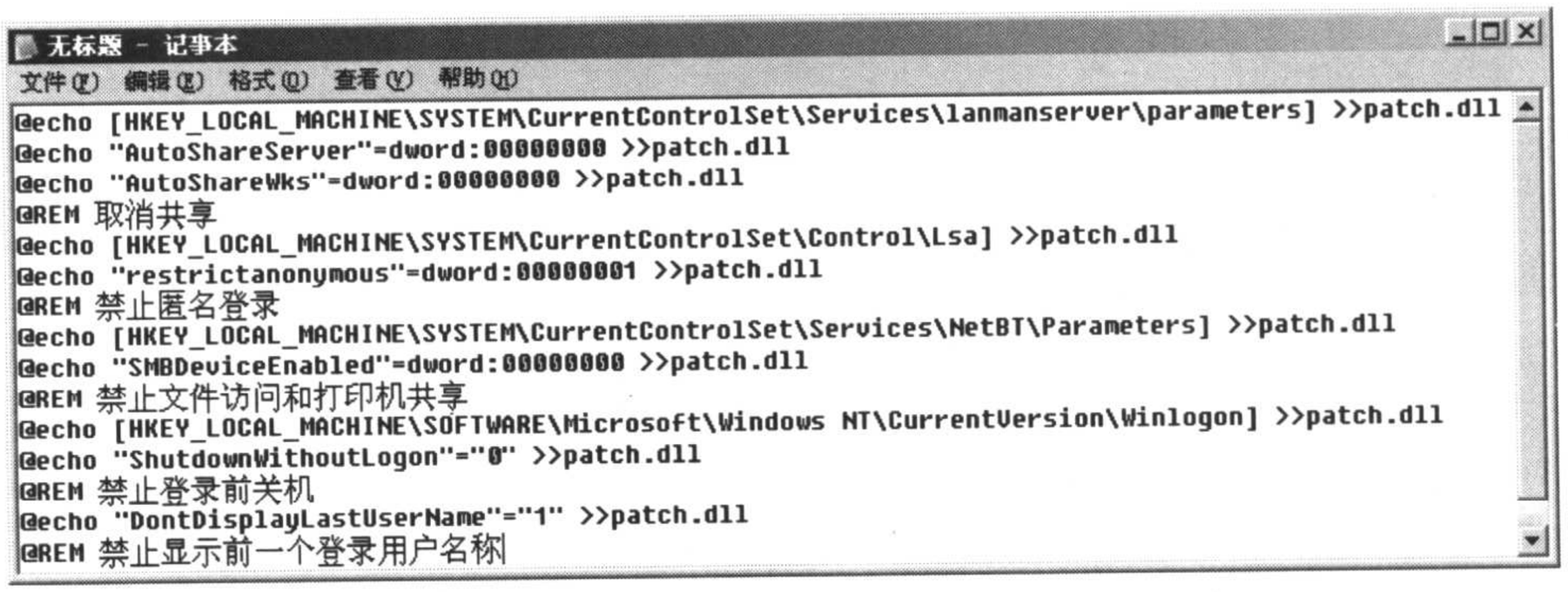

图 11-26

11.4.2　删除日志

入侵的最后一个步骤是脚印的擦除，清除日志中的内容是必不可少的。使用批处理文件清除日志可以达到事半功倍的效果，以下的代码的功能就是删除日志文件。

@net stop w3svc

```
@net stop event log
@del C:\winnt\system32\logfiles\w3svc1\*.* /f /q
@del C:\winnt\system32\logfiles\w3svc2\*.* /f /q
@del C:\winnt\system32\config\*.event /f /q
@del C:\winnt\system32dtclog\*.* /f /q
@del C:\winnt\*.txt /f /q
@del C:\winnt\*.log /f /q
@net start w3svc
@net start event log
```

代码中首先关闭了日志记录功能，然后对日志进行了清除，清除完毕后，重新打开了日志记录功能。

11.5 小结

本章介绍了批处理文件中的常用命令及批处理文件的编写方法，最后给出了两个综合利用的实例。批处理文件的相关知识虽不是入侵过程中需要必备的知识，但较好地掌握批处理文件将会大大提高入侵的效率。

目录
正文